FLIGHT SYSTEM TESTBED
FUNCTIONAL CAPABILITIES

Rapid spacecraft prototyping for early, cost-effective subsystem integration, system-level interface compatibility, and system performance verification

---

Jet Propulsion Laboratory
California Institute of Technology

JPL D-12025 7/95

PREFACE

The Jet Propulsion Laboratory (JPL) of the California Institute of Technology has over 30 years of experience in developing spacecraft and managing deep space missions. Because of the success of these deep space missions, JPL is considered a world leader in space technology.

In response to the National Aeronautics and Space Administration's (NASA's) goal of "faster, better, cheaper," JPL has developed extensive plans to minimize cost, maximize customer and employee satisfaction, and implement small and moderate-size missions. These plans include improved management structures and processes, enhanced technical design processes, the incorporation of new technology, and the development of more economical space and ground system designs. The Laboratory's new Project Design and Strategic Planning Office has been chartered to oversee these innovations as well as the reengineering of JPL's project design process, including establishment of the Project Design Center (PDC) and the Flight System Testbed (FST).

The FST comprises a group of test sets that include a series of project-specific testbeds and a permanent JPL testbed to be used for technology infusion. As new technology is accepted, it will be infused into project-specific testbeds, producing an evolving body of knowledge that will be accessible to flight projects. With this innovative approach, cost and risk are reduced, problems can be solved prior to costly flight qualification, and advanced technology can be built into the spacecraft design much earlier and with greater confidence than with conventional methods.

CONTENTS

FIGURES

1
Introduction

In the early 1960s, when JPL first embarked on the task of planetary exploration for NASA, little was known about the type of spacecraft required or the environment the spacecraft would encounter. Funding was generally not a limiting resource --- the biggest challenges were the technical issues. In those days, the Laboratory usually built three spacecraft --- two to fly and one as a proof test model (PTM). The PTM was built first, and only after thorough integration and testing was the design finalized and work begun on the two “flight” spacecraft. The PTM was usually constructed of non-space-qualified components that were equivalent in form and function to the components that were flown later; they provided the basis for the assembly, test, and integration of the two flight spacecraft. Although costly, this concept allowed design and test issues to be resolved early and ensured that the flight spacecraft would be completed in time to meet the scheduled, inflexible launch date.

As funding became limited with the development of larger and more sophisticated spacecraft, JPL reevaluated its spacecraft-development process. The confidence gained from many years of building planetary spacecraft allowed the Laboratory to eliminate the PTM and build only the two flight spacecraft. Although this approach saved money, most of the system-level problems emerged late in the development cycle, adding more risk to the schedule --- which usually required cost reserves to meet a launch date. Later, the cost of flying two spacecraft became prohibitive, and only one could be justified. This additional constraint caused JPL to perform a higher level of analytical modeling to ensure system performance and reduce the risk of a single-point failure.

In the 1990s and in the 21st century, JPL faces new challenges in accomplishing one of its strategic goals: to define, develop, and implement a series of scientifically exciting, publicly engaging, and financially affordable, small and moderate-size missions to meet NASA’s planetary exploration goals. To assist in meeting these new challenges, JPL has established the Flight System Testbed (FST). The FST will enable the Laboratory to develop small, high-technology, low-cost spacecraft in an environment where rapid prototyping of future spacecraft architectures and designs can be accomplished in an end-to-end mission operations system (EEMOS) environment.

This rapid prototype development approach will enable JPL to:

-- Reduce project costs and shorten schedules.
-- Facilitate system and subsystem inheritance from mission to mission.
-- Evaluate new technologies and integrate them into future missions.
-- Develop flight and ground systems concurrently.
-- Evolve testbed capabilities for future mission requirements.

These are the goals of the FST.

1.1 Purpose

The purpose of this document is to provide a description of the FST's functional capabilities and support services.

The FST provides capabilities similar to those originally provided by the PTM. The FST environment allows rapid prototyping of spacecraft architectures and designs to be completed early in the project life cycle when they are most cost-effective. The FST's spacecraft-prototyping focus is on the early resolution of individual subsystem integration, system performance verification, and system-level interface compatibility; generally it does not involve the spacecraft's mechanical structure. Early prototyping can be accomplished with various high-fidelity simulated subsystems that are interchangeable with the hardware-prototype subsystem models.

1.2 Scope

This document describes the current FST capabilities as well as the plans for enhancement of those capabilities. The FST organizational structure and facility layout are included to provide the reader with implementation and operations insight into the development of the FST.

1.3 Document Organization

Section 2 describes the FST functional organization, facility, capabilities, and connectivity. The approach to new technology infusion is also described. Section 3 discusses FST subsystem configurations and capabilities. It is expected that this document will be updated at least annually to reflect enhancements; additionally, interim revisions will be made whenever there is a major change in FST capability.

2
The Flight System Testbed

To fulfill the goals delineated in Section 1, the FST will assist users in reaching these objectives:

-- Identifying at an early stage spacecraft and EEMOS system-level interface and architecture problems.
-- Reducing time and cost of system integration by providing a “pathfinder” effort.
-- Evolving towards standardization of system-level interfaces.
-- Supporting concurrent flight and ground system engineering and development.
-- Transferring technology to future projects.
-- Providing spacecraft and EEMOS design continuity from project to project.
-- Providing pre- and postlaunch sequencing and flight software verification capabilities.

To meet these objectives, a new approach to spacecraft design and development is needed. This approach requires a centralized, flexible, modular capability for developing and evaluating spacecraft prototypes and associated ground data systems. Centralization provides for capabilities to be either located in one area or remotely connectable to the FST. Flexibility implies the use of both hardware components and software simulations. Modularity allows for the integration of alternate implementations of systems, subsystems, and components.

2.1 Organization

The FST Program Office has organized around the processes that meet the goals and objectives of the FST. The organization (shown in Figure 1) has four distinct functional areas supported by the FST manager and system engineering staff.

The FST functional areas provide the following support:

-- Advanced Planning and Technology Evaluation plans for future capabilities, evaluates new technologies, and provides for the integration of new technology into the FST prototyping environment.

-- Facility and Infrastructure Development designs and implements the facility hardware and software capabilities necessary to support the other functional areas within the FST.

-- Project Integration and Test implements project-specified and specific prototyping activities within the FST and maintains the FST virtual spacecraft prototype and related technology integration activities.

-- Simulation Development designs, implements, integrates, and tests spacecraft and ground subsystem simulators.

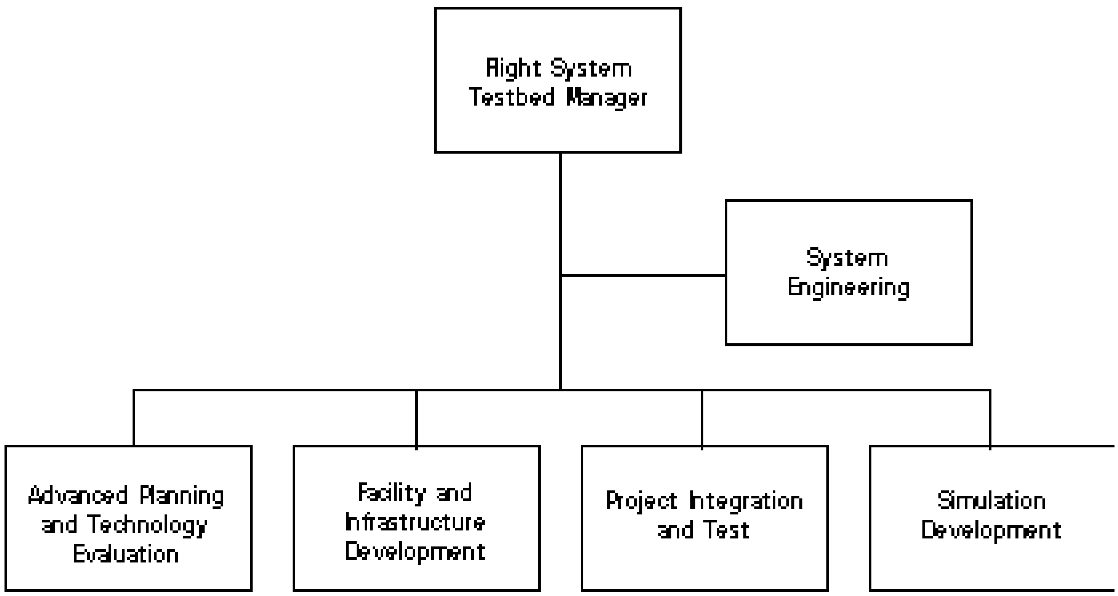

Figure 1. FST Functional Organization.

2.2 Facility

The FST has been designed to operate as a central facility with the ability to expand to multiple remote facilities as necessitated by user requirements.

The FST's primary location is in Building 179 at the main JPL site on Oak Grove Drive in Pasadena, California. The central FST facility (illustrated in Figure 2) contains three test sets with supporting equipment, which are dedicated to FST integration and test activities. There is a presentation area for technical briefings and user meetings, offering video displays, overhead projection, and color projection viewing for computer-driven images and text. The shared work area for FST programmers and engineers contains UNIX workstations, IBM-compatible personal computers (PCs), and Macintosh computers. A technical support laboratory (cabling area) is used to perform small maintenance tasks or modifications to systems under test.

2.3 Capabilities

A typical spacecraft is composed of individual subsystems --- command and data handling (C&DH), telecommunications (telecom), the attitude and articulation control subsystem (AACS), and power (PWR), along with instrument payload(s). A ground data system (GDS) is provided to allow uplink commanding and downlink telemetry and science.

The FST is developing generic simulation models of individual spacecraft subsystems that can be customized to meet the specific requirements of a given spacecraft design (see Figure 3). Furthermore, the FST treats the payload as another subsystem and is developing an instrument simulation feature that can emulate one or more payloads, as required by the mission. A Spacecraft Dynamics Simulator (SDS), composed of software models, provides spacecraft sensor data to the AACS based on thruster firings and other events that affect spacecraft orientation.

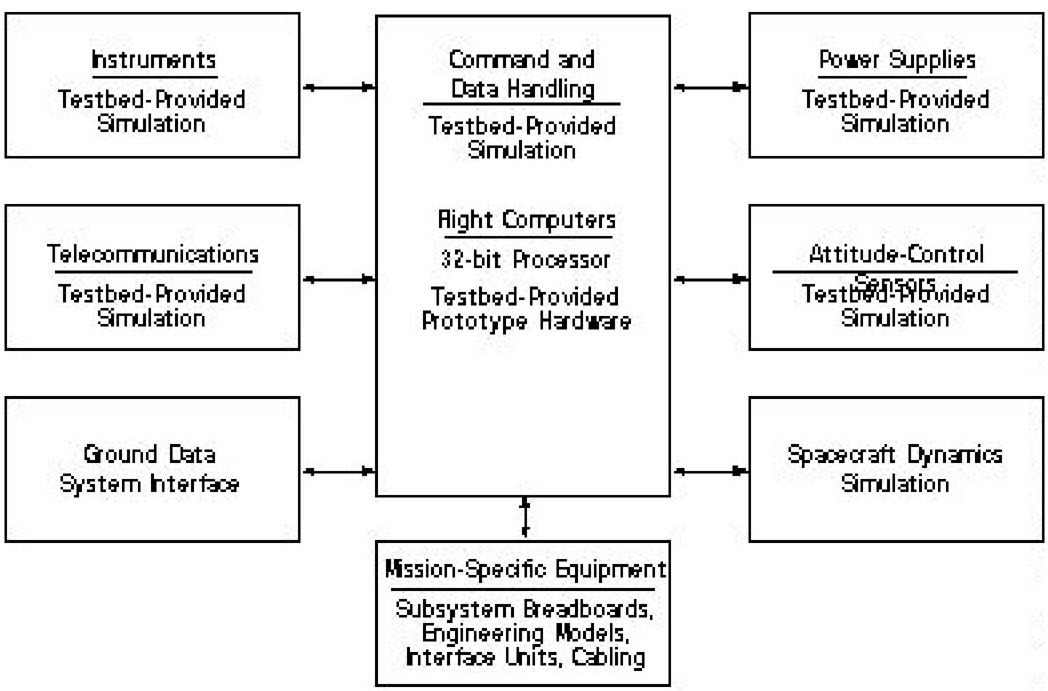

Figure 3. FST Simulation Models.

Typically, hardware models of spacecraft subsystems are developed in three phases. First, the subsystem components are delivered as a breadboard that is functionally equivalent to the flight unit but is usually different in form and size. The breadboard is then replaced by the engineering model, which is equivalent in form and function to the flight unit. The flight unit will replace the engineering model in the final configuration. The unique FST subsystem simulations can be tailored to resemble a specific subsystem, thus allowing a spacecraft to be designed from simulated, functionally equivalent subsystems. This virtual (simulated) spacecraft approach permits early validation of the system design and intersubsystem interfaces. As the hardware models for the individual subsystems arrive, they can be substituted for the simulations, and integration and testing can continue until all the simulations and breadboards have been replaced with engineering models (Figure 4). If problems arise with a component or subsystem, the system integration and test can still continue using simulations or hardware, as necessary. The FST provides a virtual spacecraft composed of hardware and simulated subsystems, which forms the genesis for a pre- and postlaunch flight software and sequencing verification system.

Within the FST, a virtual spacecraft can be readily configured for a specific mission in order to evaluate new or alternative technologies. This rapid prototyping approach enables system-level problems to be identified and resolved early, making the development process faster and more cost-effective. As the FST evolves, it will permit the creation of independent, virtual spacecraft configurations for each unique mission type. Future projects will be able to take advantage of this capability to reuse existing spacecraft hardware and software.

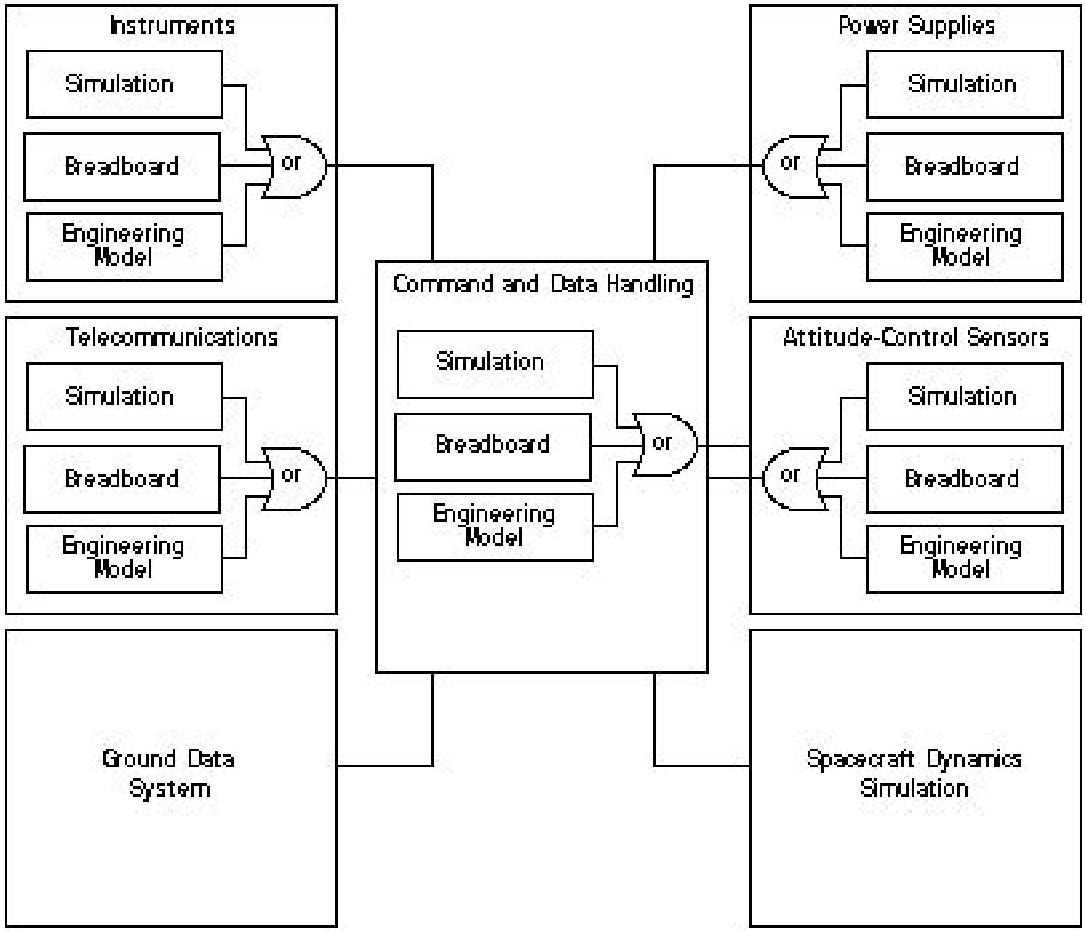

Figure 4. Configuration Options.

There are significant cost and schedule benefits associated with developing standardized system-level interfaces and providing system design continuity from current to future missions. With standardized system-level interfaces, integration of the various components and subsystems is no longer a major technical concern, and less time can be spent on the design and testing of the interfaces. Design continuity allows for easy prototyping implementation and trade-off analyses of incremental changes to the design, which are necessary to meet the requirements of future missions. It should be noted that the term “system design” normally refers to the spacecraft system; however, the FST incorporates elements of the GDS, so the entire EEMOS can be represented. Because both the ground and flight elements of the EEMOS are available in the FST, design changes and trade-offs are not limited to the spacecraft --- thus providing a concurrent development environment for the entire EEMOS. This cost-effective approach can streamline EEMOS integration and test by allowing early verification and validation of system test procedures.

The GDS provides the spacecraft commanding and telemetry-processing capabilities of the mission operations system (MOS). This allows the integration of an integral part of the MOS into the design process so that future flight development can include ground requirements, thus avoiding costly and time-consuming rework of the ground-systems facilities. This early integration of the flight and ground systems facilitates concurrent development of these systems through identifying incompatibilities and bottlenecks, prototyping architectural trade-offs, and analyzing the results.

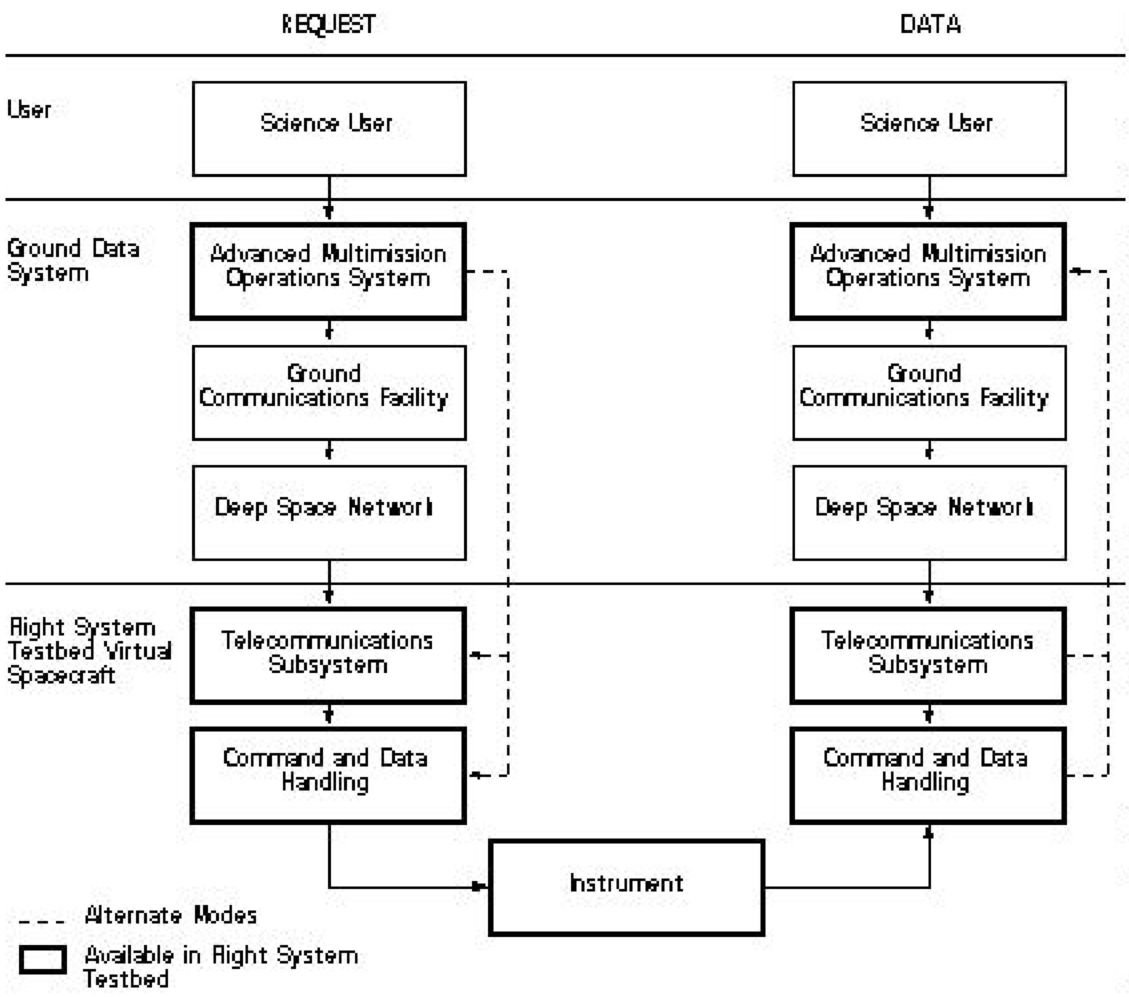

Figure 5. End-To-End Data Flow.

The typical end-to-end data flow (Figure 5) begins with a request by the user of the system and ends with data satisfying the user's request. The path of the request and data flow is common to almost all missions. The GDS processes and forwards the request to the spacecraft instruments for science data, and the instruments return the requested data, which are then collected and sent by the spacecraft to the ground. The GDS then processes and distributes the data to the user. If users wish to choose an alternate mode of connectivity, they can bypass those parts of the EEMOS (see alternate mode paths in Figure 5) that are not necessary in the current prototyping activity.

2.4 Connectivity

The FST is linked to a transmission control protocol/Internet protocol (TCP/IP) Ethernet local area network (LAN) that is accessible from remote locations at JPL and at other NASA Centers, universities, and industry (Figure 6). This interconnectivity allows convenient electronic access to components and subsystems that are required for testbed activities. The FST is currently extending remote connectivity to a laboratory involved in determining the structural effects on large spacecraft bodies, specifically instrument booms, that result from changes in temperature and movement. This is the start of an astrophysics testbed and will be used to tie the interaction of these large structures to the spacecraft AACS in the FST.

2.5 New Technology in the FST

The evaluation of new technologies and their incorporation into future missions will enable the Laboratory to continue to build the best spacecraft possible and return the highest quality science available. JPL's FST will foster cooperation with other NASA Centers and industry to identify new technology candidates for evaluation in the FST. As the new technologies are evaluated, they will be integrated into current spacecraft designs, thus allowing these new technologies to be evaluated in a system context (Figure 7).

The technology-infusion effort in the FST provides a system-level venue for technology developers to demonstrate to mission users the operability and functionality of these new technologies. The system-level venue is the EEMOS, which is composed of modular components and standard interfaces allowing easy integration and evaluation of new technologies.

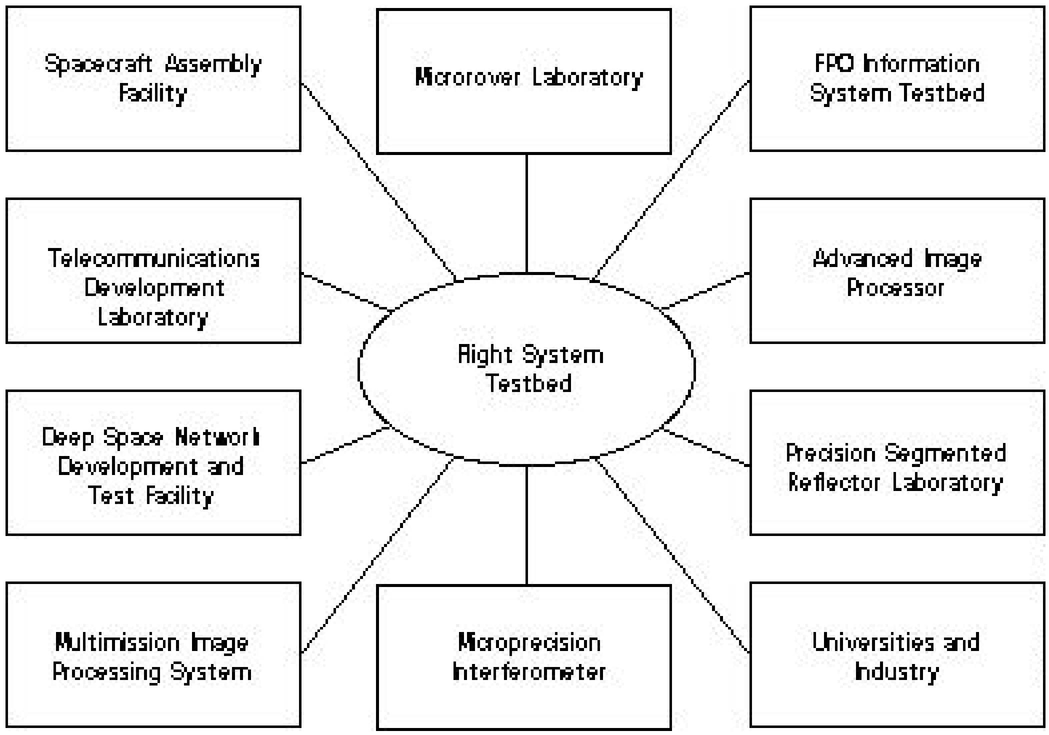

Figure 6. FST Remote Connections.

2.6 FST End-To-End Architecture Demonstration

The FST has an end-to-end demonstration of its architecture that includes elements of the flight system, payloads, support equipment, and GDS. Whenever possible, products brought to the FST for evaluation will be integrated into this demonstration. This provides a consistent scenario across products, makes the maximum use of existing FST assets, and demonstrates the applicability of the product to projects in a system context. The core of the demonstration is a hardware-in-the-loop, real-time simulation that includes uplink and downlink processes for data recovery and mission operations. The spacecraft simulation can be flown through maneuvers, including capture of simulated science images and data tied to the state of the simulation. Thus, there exists a place in this simulation to host nearly any type of product.

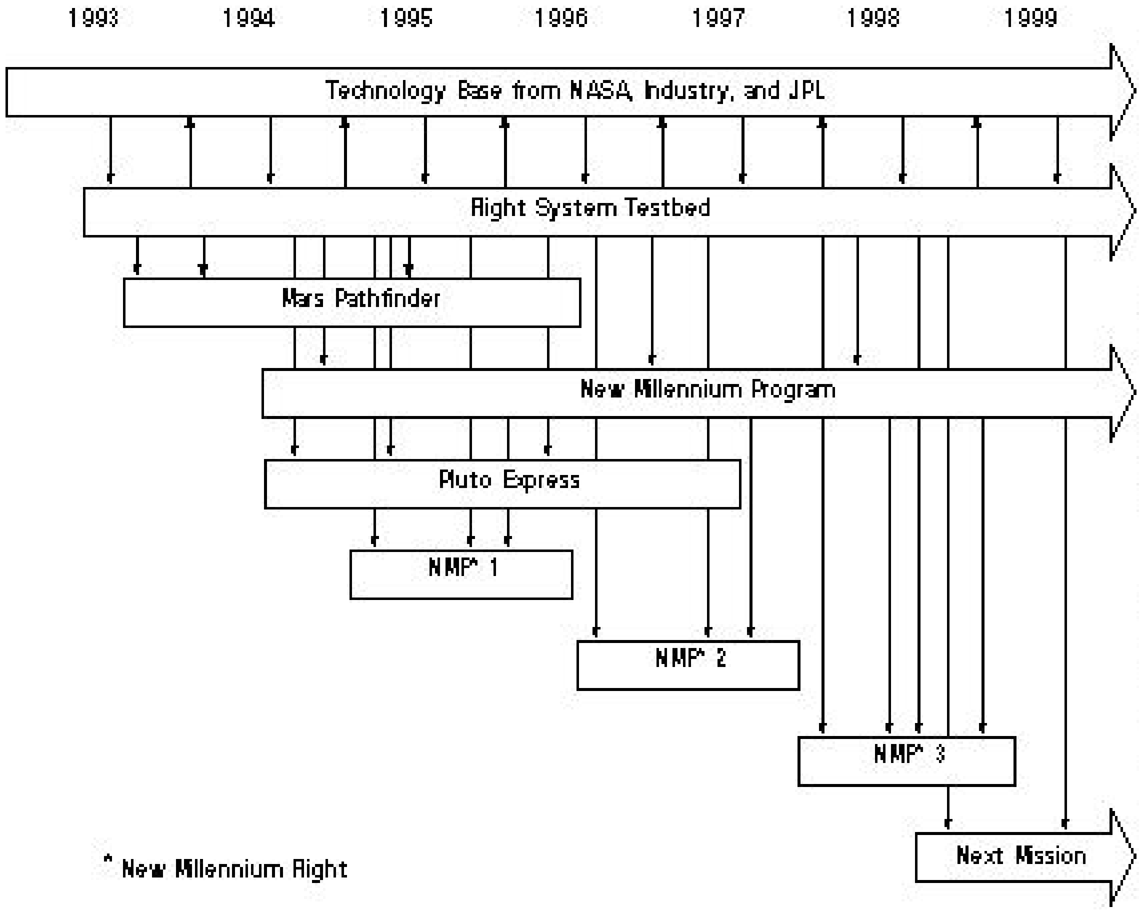

Figure 7. Technology Infusion.

The demonstration utilizes several elements, as illustrated in Figure 8. The JPL Multimission Ground Data System (MGDS) software is used to capture and display downlink telemetry and to prepare and send uplink commands. The support equipment transfers these functions to the spacecraft electronics by means that more or less faithfully simulate the Deep Space Network (DSN) processes and telecommunications subsystem. A telecommunications subsystem simulator is under construction that will provide a flight-like interface to the spacecraft and a socket interface to the support equipment.

The spacecraft electronic data system currently contains a few simple elements. C&DH is modeled as a simple switch that routes data streams between onboard devices. One such device is the solid-state recorder (SSR), built from a large computer memory block used as a named file system. Another device is the remote engineering unit (REU), which samples 10 potentiometers with an analog-to-digital (A/D) converter to provide either simulated engineering telemetry or science data in real time.

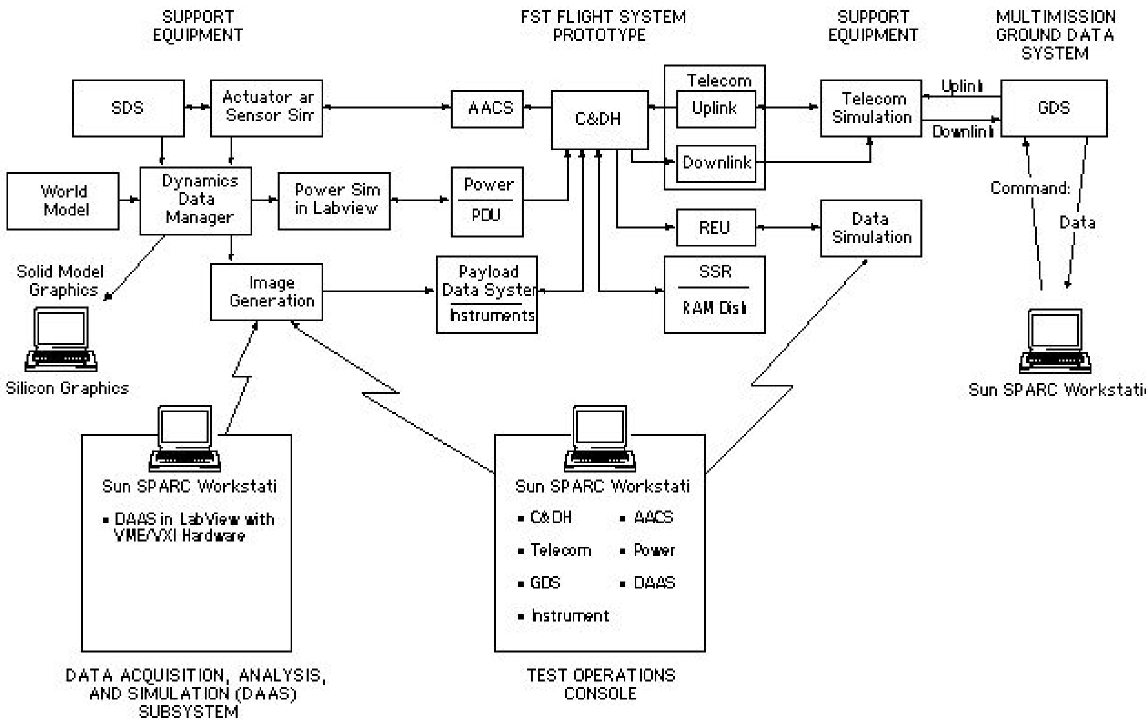

Figure 8. FST System Architecture.

The AACS is another spacecraft device addressable from the C&DH subsystem. The attitude controller consists of a commander that generates the turn profile and a controller that follows the profile. Initially, perfect knowledge of attitude and rates is assumed, but sensor models will be incorporated in the near future. The simulation uses thrusters to control the spacecraft, but other devices will be added.

The simulation is based upon real-time integration of the equations of motion for flexible spacecraft. For most small spacecraft missions, rigid-body mechanics should be sufficient, adding extra bodies to account for reaction wheels or scan platforms as appropriate. The real-time simulation can handle at least five bodies at a 25-Hz rate. For rigid bodies, the simulation throughput will probably be limited by the computations in the sensor and actuator dynamics rather than the flight mechanics. Depiction of the simulation in solid-model graphics in real time is also available.
Attitude-control loops are closed through sensor simulations (or sensor electronics, if available) through the support equipment. The real-time dynamics solution is used to provide position and rate truth knowledge to the sensor simulations. Generation of synthetic image scenes for cameras is entered through instrument simulation or into the camera electronics.

3 Subsystem Capabilities

The FST is designed to support project system-level integration and test using simulated and real spacecraft and ground subsystems (refer to Section 2.3). Five spacecraft subsystems (excluding mechanical) --- C&DH, telecom, instrument payload, AACS, and power --- represent the spacecraft and are supported by four support subsystems --- SDS; Data Acquisition, Analysis, and Simulation (DAAS); Test Operations Console; and MGDS (see Figure 9).

Testing and integration of simulated spacecraft subsystems and components provide insight into spacecraft system performance prior to integration of the hardware components of the spacecraft subsystem. As breadboards become available, they can be integrated, replacing the simulations, and testing can proceed using a combination of hardware and simulations. Breadboards can then be replaced with engineering models or flight components until the final spacecraft hardware configuration is established. Thus, there are a multitude of spacecraft configurations that can be used to provide the early testing and verification of system performance.

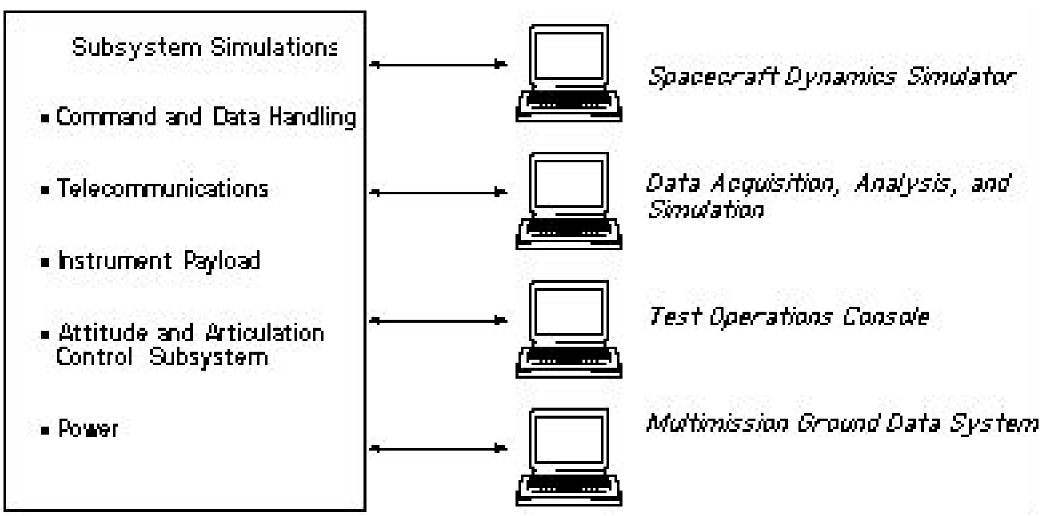

Figure 9. FST Subsystem Configuration.

The FST emphasis is on the integration of subsystem capabilities to form a complete system in which spacecraft and EEMOS designs can be evaluated for performance, throughput, and compatibility. The strength of the FST is the ability to easily interchange subsystem implementations to emulate a new design or system architecture. The ease with which an alternate subsystem implementation can be integrated into an existing spacecraft design lies mainly in the interfaces between subsystems, i.e., intersubsystem interfaces. If these intersubsystem interfaces are standard, both electrically and mechanically, then integration of the system becomes much easier. It is not proposed that all interfaces be the same, but only that as many interconnections as possible are implemented using standardized interfaces. The FST has chosen the following interface standards to implement in the subsystem simulations:

Backplane --- VersaModule Eurocard (VME)
LAN --- 1553B, 1773, Small Computer System Interface (SCSI)
Point-to-Point (Pt-Pt) --- RS232, RS422

The FST will remain flexible and add new standards as they become available and prove appropriate for the needs of future missions.

Subsystems and components with nonstandard interfaces can and will be integrated into the FST when requested by a user or project for a specific prototyping activity. The integration of these nonstandard interfaces, although costly, can be done using field programmable gate array (FPGA) technology to develop input/output boards that are standard on the spacecraft side and custom on the component side.

The remainder of this section details the specific capabilities of the five spacecraft subsystem simulations and the four support subsystems.

3.1 Command and Data Handling Subsystem Simulation

The C&DH subsystem is the central computing and data processor for the virtual spacecraft. The C&DH simulation includes a central processing computer and a data storage capability, such as the SSR. The C&DH subsystem receives, validates, decodes, and processes commands for other spacecraft subsystems, and formats and prepares commands for time-tagged execution and/or controlled timing events. The C&DH subsystem also collects, processes, and formats spacecraft housekeeping and payload data for downlink transmission. Functions performed by the C&DH subsystem are shown in Figure 10.

Uplink commands from the telecommunications subsystem are input in the form of Consultative Committee for Space Data Systems (CCSDS) command link transmission units (CLTUs) to the command decoder, which extracts code blocks from the CLTUs. The code blocks are passed to a deblocker, which decodes them to reconstitute the original commands sent to the ground command system. The decoded commands then go to the command receiver where they are routed to the appropriate onboard devices and instruments, including a sequence manager, which is responsible for executing timed and/or event-driven command sequences.

Devices and instruments produce CCSDS telemetry packets at appropriate intervals and send those packets to the telemetry transmitter and/or to the SSR for storage. Upon command from the ground station or by a sequenced command, the data from the SSR are transferred to the telemetry master channel manager. The telemetry master channel manager merges real-time telemetry from the telemetry transmitter with playback telemetry from the SSR, inserting null packets as necessary to avoid loss of sync in the downlink stream; it packages all these packets into CCSDS telemetry frames and passes the frames to a Reed-Solomon encoder. The Reed-Solomon encoder encodes the frames and passes the resulting blocks to the telecommunications subsystem for downlink.

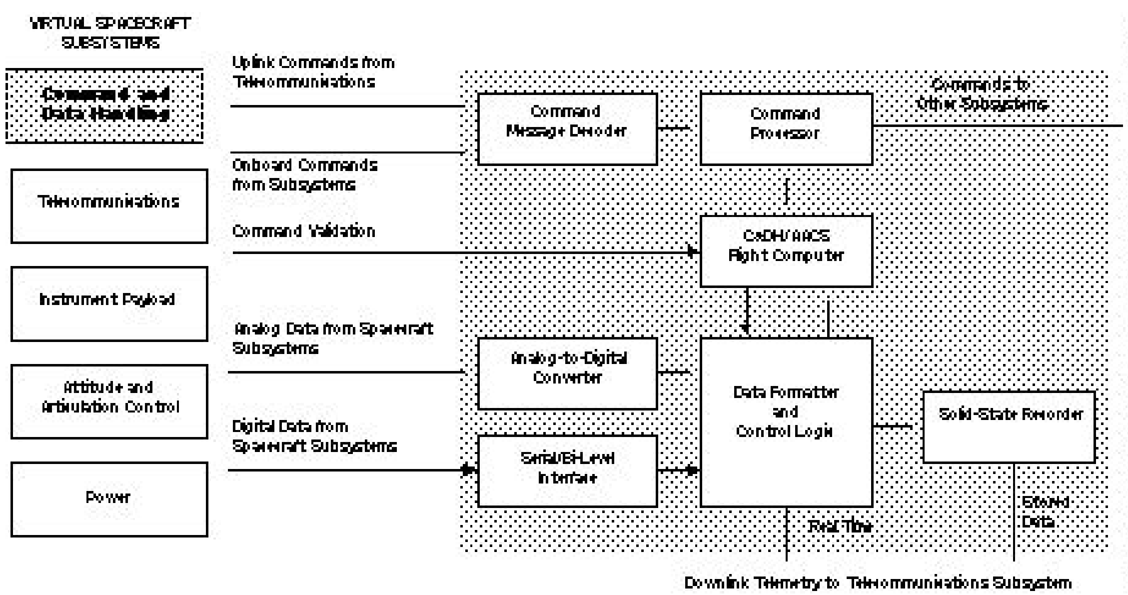

Figure 10. Command and Data Handling Subsystem.

The C&DH functions are currently implemented in a 32-bit embedded central processing unit (CPU), M68040 or R3000, which resides on a VME backplane along with other processor and input/out (I/O) boards required by the simulation. This subsystem simulation, as with the others, is implemented using commercial off-the-shelf (COTS) hardware and software, where possible, and a high-level programming language with a mature development environment. The embedded CPU uses a real-time COTS operating system, VxWorks, and the simulation is written in C.

3.2 Telecommunications Subsystem Simulation

The spacecraft telecommunications (telecom) subsystem provides the communication link between the ground and the spacecraft. The deep space transponder (DST) includes a receiver and transmitter. The command detection unit (CDU) detects the uplink command stream, and the telemetry modulation unit (TMU) modulates the downlink carrier with spacecraft telemetry data. The telecommunications subsystem is shown in Figure 11.

The telecommunications subsystem is designed to simulate the operation of spacecraft telecommunications hardware and to provide an active interface between the C&DH subsystem and the GDS. The FST's radio frequency simulator (RFS) software can directly model the physical elements of the actual telecommunications subsystems. The RFS subsystem consists of three main components: the RFS model (which simulates the RFS hardware), the RFS interface element of the spacecraft's flight software, and a direct user interface to the RFS model. The RFS model runs under the VxWorks operating system using an embedded single-board computer (SBC). It comprises the following:

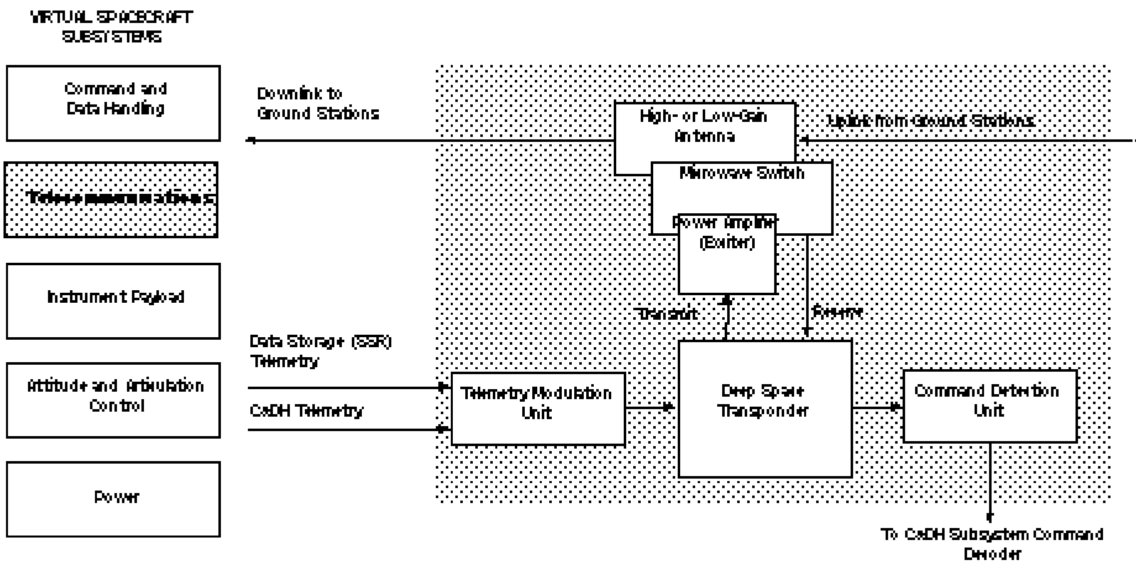

Figure 11. Telecommunications Subsystem.

-- A C&DH interface object, which manages the state and configuration of the RFS model as a whole.
-- Up to two transponder (DST) objects.
-- Up to two decoder (CDU) objects.
-- Up to two switch objects.
-- Up to two modulator (TMU) objects.
-- Up to two RF amplifier objects.
-- Up to five high- and/or low-gain antenna objects.

The flight software RFS interface accepts commands from the C&DH subsystem and transforms them into RFS subsystem configuration directives, which are processed by the RFS model's C&DH interface object. It also receives RFS subsystem status data from the model's C&DH interface and uses that data to build and issue RFS state telemetry packets.

The user interface program --- i.e., RFS support equipment --- runs on a Sun SPARC workstation under UNIX and provides the following:

-- A text-based interface that accepts RFS subsystem configuration directives entered as ASCII text. The response or current state of each object as it changes is printed to the display.

-- A graphical user interface (based on X and Motif) that translates mouse button clicks into directives and then displays the simulator state in a combination of text and graphic presentations.

The user interface program and the flight software RFS interface both send directives to the RFS and receive acknowledgments and responses from the C&DH interface object via a TCP/IP socket interface. Communication among the RFS model's objects is by message queues and named pipes. The DST objects communicate with simulated command sources and telemetry links via TCP/IP sockets. The TMUs and CDUs communicate with the C&DH uplink and downlink elements via an RS232 serial interface.

The RFS is controlled by the user interface program or by commands issued from the ground. RFS configuration directives written in a simple syntax provide operators with the ability to:

-- Change the value of one or more attributes of a simulation element.
-- Toggle the "watch" state of a simulation element between on and off. When "watch" is turned on for a simulation element, every change in the value of any attribute of that element will always cause the complete status of the element to be reported to all user interface processes connected to the simulator.
-- Query the values of the attributes of a single simulation element or all simulation elements.
-- Turn on or off a given simulation element.

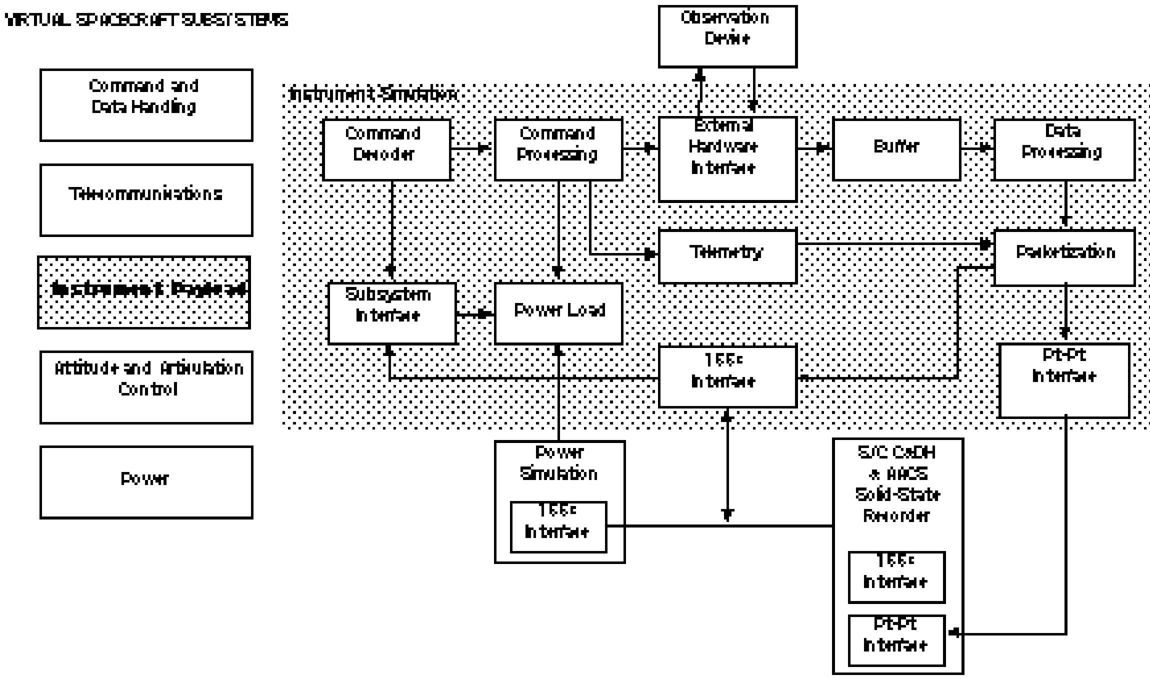

Figure 12. Instrument Subsystems.

3.3 Instrument Payload Subsystem Simulation

The simulation of instrument subsystems in the FST can be customized to meet the payload requirements of a specified mission. This capability is not limited to a single simulation at a time; if necessary, the FST can support many simulations. The simulation provides a high-fidelity, system-level interface that can be tailored to meet the software protocol and hardware interface specifications of the actual payload instruments. The simulation also provides an interface for a real hardware instrument that may not have the capability to communicate with the other spacecraft subsystems.

The instrument subsystem simulation (Figure 12) runs in an embedded 32-bit processor under a real-time operating system with the necessary I/O devices for communication with the other subsystems. The instrument subsystem simulation interface to the typical spacecraft, as controlled through the FST, can be via spacecraft networks (buses), a direct point-to-point connection, or a combination of both. The bus interfaces currently being developed include the VME backplane, as well as the MIL-STD-1553B and 1773 LANs; the point-to-point serial interfaces are RS232 and RS422.

The instrument subsystem simulation contains numerous processing tasks that surround the actual data simulation task. The input processing includes handling, decoding, and processing the commands from the C&DH subsystem; the output processing includes buffering and packetization of the data. The data are then sent to the appropriate subsystem via the spacecraft bus or a direct connection. An interface to the power subsystem is included for a sinking that is equivalent to the specified operating modes of the instrument.

The instrument subsystem simulation can be used to “front-end” an instrument for integration into the spacecraft design. For example, a dedicated observation device may have no way of communicating with the other spacecraft subsystems. However, if this device is required for a future mission, the FST can simulate an interface to it. This allows the rest of the spacecraft to interface normally with the instrument and at the same time receive real data from the instrument instead of data simulated by an analytical model.

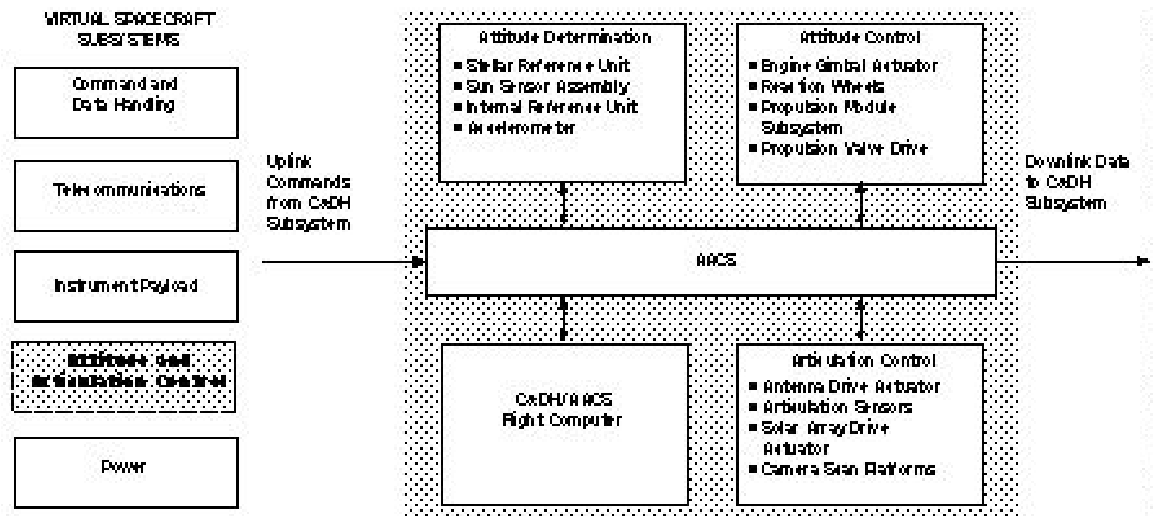

Figure 13. Attitude and Articulation Control Subsystem.

3.4 Attitude and Articulation Control Subsystem Simulation

The AACS provides the control capability to autonomously maintain attitude orientation and stability of the spacecraft during all phases of a mission. The AACS also provides the control capability to support the required AACS functions associated with communications, power, thermal control, and propulsion maneuvers. Figure 13 shows an expanded view of AACS functions.

Spacecraft attitude changes and the pointing of instruments, cameras, or other spacecraft devices are performed by the AACS. Additionally, guidance and navigation commanding, control, and housekeeping are performed by the AACS. The attitude-determination subsystem provides reference information on the spacecraft attitude; the attitude-control subsystem controls the attitude during cruise mode and during maneuvers of the spacecraft. Articulation control manages the attitude of onboard devices (camera pointing, solar array position, antenna pointing, etc.).

The AACS simulator is addressable from the C&DH subsystem. The attitude-control system includes a commander that generates the turn profile and a controller that follows the profile. The control system monitors and commands the simulated spacecraft via a 1553 bus or via a simulated bus. The spacecraft simulation (Dshell/DARTS) interprets the commands to effect a change in the state of the spacecraft (e.g., a turn).

A library of attitude-control sensors and actuator models is being collected so that design against specific spacecraft configurations can be accomplished easily and quickly. The library models are being implemented to be easily adapted to future mission sensor and actuator requirements.

3.5 Power Subsystem Simulation

The power subsystem simulator supplies, controls, regulates, and distributes all electrical power required for the spacecraft. This capability is required for the spacecraft continuously --- during integration, prelaunch testing, and the transition to spacecraft internal power during the launch phase, as well as all subsequent mission phases. A typical power subsystem is shown in Figure 14.

The power subsystem simulator provides DC power to spacecraft components and subsystems through a computer-controlled environment. It also provides separate regulated power output channels simultaneously to a variety of spacecraft loads. The power distribution unit (PDU) distributes power to the virtual spacecraft from DC power supplies that can be set to plus or minus output voltages ranging from 0 to 60 volts, with a maximum power output capacity of 1,000 watts.

Current and voltage sensing are done in the PDU by the power system controller (PSC). The PSC is made of the Sun SPARC 10 computer, VXI data acquisition, and control instruments. The instrument control program is written in LabView. The status of the PDU is constantly sensed and subsequently scanned by the PSC. The power subsystem provides telemetry (voltage, current, and PDU status) to CDS through an Ethernet socket.

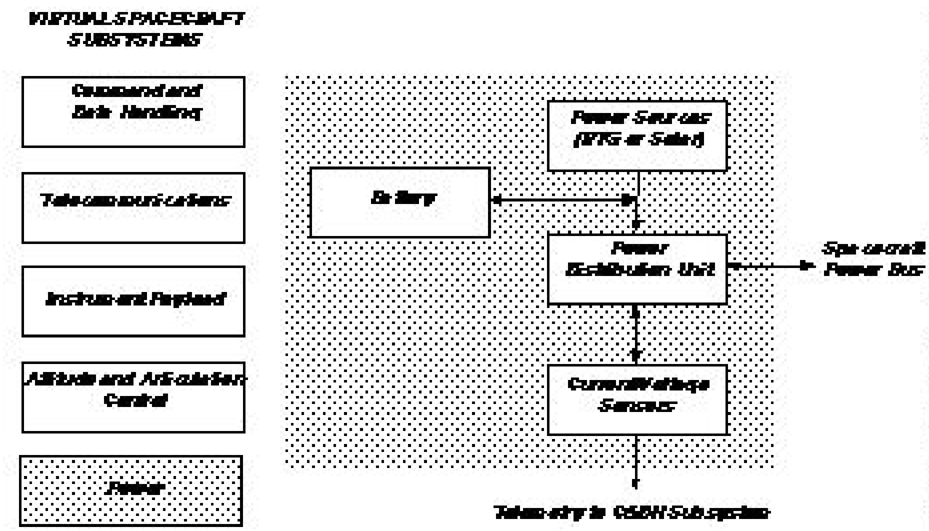

Figure 14. Power Subsystem.

3.6 Spacecraft Dynamics Simulation

This simulation is based on real-time integration of the equations of motion for the specific spacecraft. For most small spacecraft missions, rigid-body mechanics should be sufficient; extra masses can be added to account for reaction wheels or scan platforms as appropriate. Simulations can be viewed through a solid-model graphic in real time.

The SDS provides real-time, closed-loop simulation for the testing and verification of AACS hardware and software. The SDS is composed of three elements (see Figure 15):

-- Dynamics Algorithm for Real-Time Simulation (DARTS).
-- Dynamics Visualization Environment (DVE).
-- Dynamics Model Generator (DMG).

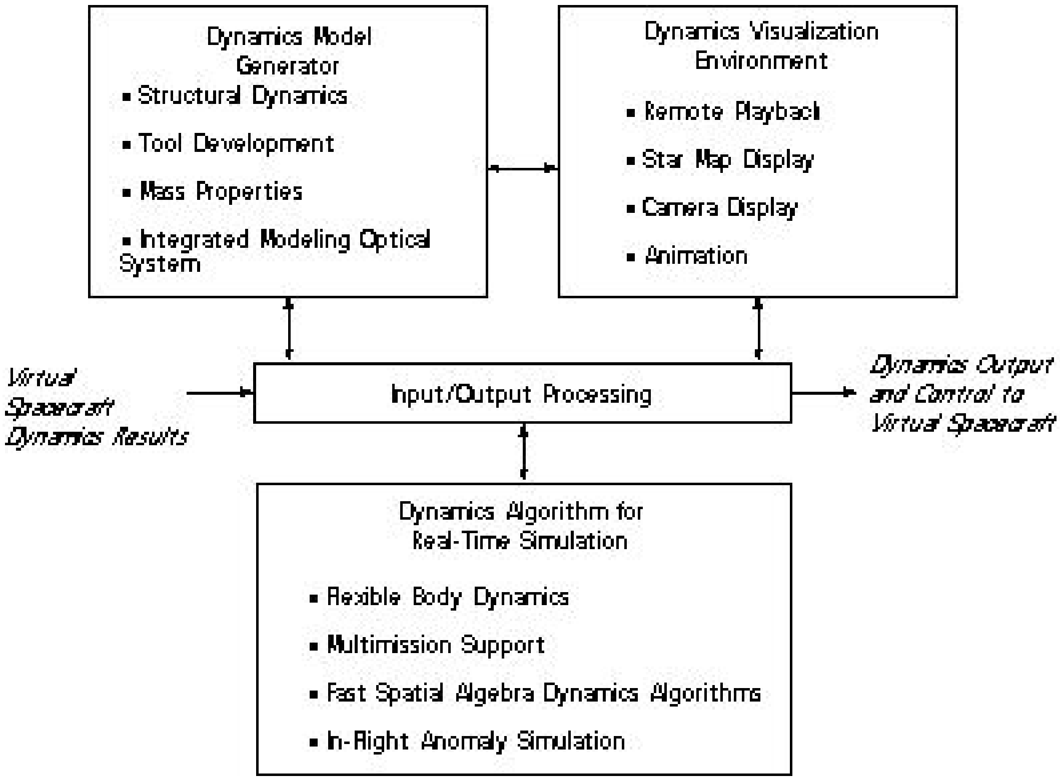

Figure 15. Spacecraft Dynamics Simulator.

3.6.1 Dynamics Algorithm for Real-Time Simulation

DARTS is a multifunction spacecraft dynamics simulator that supports:
-- A menu of selectable configurations to support multimission needs.
-- Fast spatial algebra dynamics algorithms for real-time spacecraft dynamics.
-- Flexible body dynamics.
-- A selectable quantity of actuators and sensors.
-- Prescribed motion selections at articulated degrees of freedom.
-- Data-driven control so that dynamic changes to spacecraft configuration will not require changes to the software.
-- Flexible software configurations so that DARTS can be embedded within a variety of project environments.
-- Real-time changes to the virtual spacecraft model to simulate any number of in-flight anomalies such as fuel depletion, fault insertion, probe release, fuel slosh, etc.

3.6.2 Dynamics Visualization Environment

DVE is hosted on a graphics workstation and provides enhancements to evaluate the results of visual test scripts and scenarios. Its functions include:

-- Remote, playback, and interactive operation for real-time and non-real-time operation.
-- Spacecraft animation capability.
-- Multiple port views of the spacecraft from different angles.
-- Star-map display.
-- Camera display views of star maps.
-- Simulation data graphic displays of visions in real time.
-- Camera lighting control and viewpoints of celestial images.

3.6.3 Dynamics Model Generator

The DMG module is a software-code generator that provides real-time generation of dynamic test modules for DARTS. The DMG can:

-- Accept an Integrated Modeling Optical System (IMOS) structural dynamics model for spacecraft input.
-- Develop tools to substructure the spacecraft model based on the bodies that undergo large-angle motions.
-- Generate mass properties, kinematics data, and structural flexibility data for the subelements.
-- Integrate selected model-reduction modules into the environment.
-- Develop a DARTS-compatible spacecraft dynamics module.

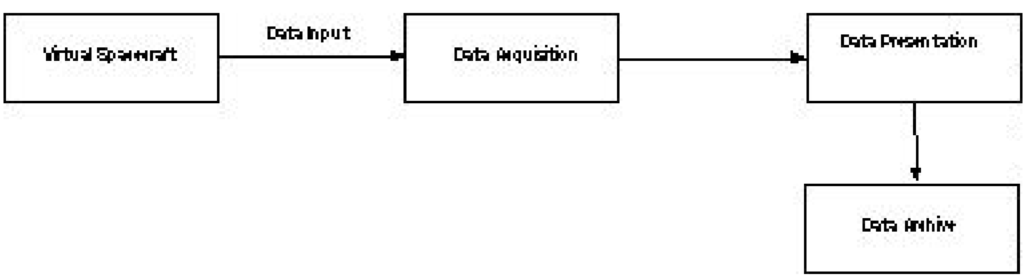

Figure 16. Data Acquisition, Analysis, and Simulation Subsystem.

3.7 Data Acquisition, Analysis, and Simulation

The FST DAAS subsystem performs data acquisition, monitoring, and analysis, and consists of a data acquisition subsystem, a data presentation subsystem, and a data archival subsystem. The FST support team will provide assistance in constructing specific analysis routines from algorithms specified by the flight project design engineers. In addition, the DAAS subsystem will provide some generic analysis routines, including tabular listings of out-of-range values, near-real-time displays, rates of data received, and volume of data. The DAAS subsystem is illustrated in Figure 16.

The basic requirement for the DAAS subsystem is to monitor system-level interfaces, i.e., interprocesses at the subsystem level for a spacecraft under development and test within the FST.

The DAAS subsystem can most accurately be described as a metatool (a tool to build tools) since it will provide generic capabilities that can be customized to meet the needs of various flight projects. The functional capabilities of the DAAS subsystem include:

-- Data monitoring. The DAAS subsystem will monitor and acquire data passing across a communication medium from one spacecraft subsystem to another. This monitoring process is unobtrusive except for the necessity of a physical tap into the communication line or bus.
-- Data display. The data-presentation subsystem will permit the collected data to be displayed in digital and/or graphical form in near real time.
-- Data time stamp. The data acquired by this subsystem will be time stamped. For example, if Ethernet packets of information are being collected, each packet will be given a time stamp.
-- Data archiving. Data with time stamp will be recorded in a data archive file for real-time and off-line analysis.
-- Multiple sources of data. The DAAS subsystem is capable of monitoring, displaying, and archiving data from multiple sources from a single virtual spacecraft.

The DAAS subsystem will have the capability of presenting data to the user in many forms. Raw or decoded data can be displayed digitally with or without associated time-stamp information. This same raw data can be displayed graphically in a line or bar graph with time as one axis. The user can choose appropriate colors for graphs to accentuate various features. The interface to this system will be menu- and window-based, supporting mouse manipulation.

The DAAS subsystem will provide the capability to generate simulated data in appropriate formats and to inject these data into the various communication lines or buses using the necessary communication protocol. Given a specific format and data rate, the DAAS subsystem will generate appropriate data in a given sequence or at random (from a specified range), and at the required rate.

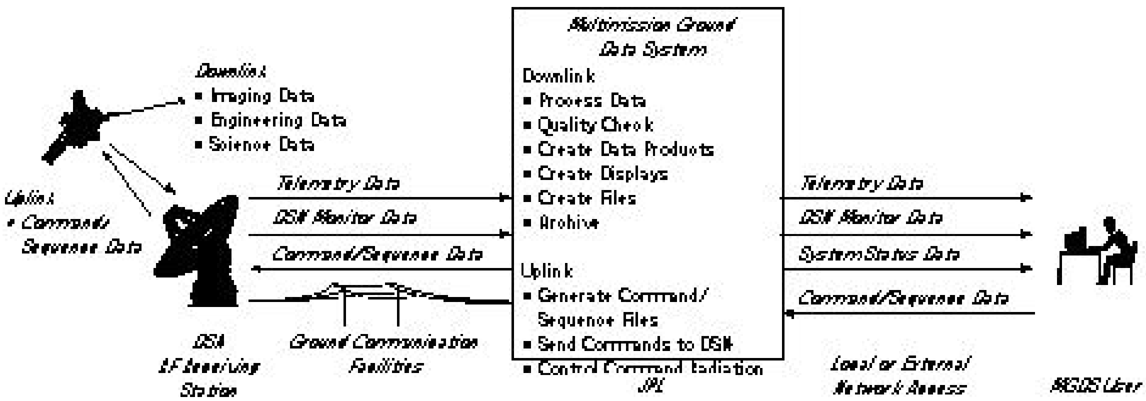

Figure 17. Multimission Ground Data System.

3.8 Multimission Ground Data System

The MGDS currently consists of the Multimission Operations Systems Office (MOSO)-provided Advanced Multimission Operations System (AMMOS). The MGDS is designed to supply a baseline set of ground data system capabilities required by all missions, with the flexibility to adapt and add new capabilities to meet additional unique requirements of individual missions. AMMOS furnishes, in a central location, a system of hardware and software components, operating teams, facilities, and procedures that work together to support JPL missions. AMMOS-provided mission control, data processing, and data storage services are vital to mission success.

The MGDS (see Figure 17) plays a key role in FST activities by providing a direct interface to the virtual spacecraft for uplink and downlink operations. A summary description of the MGDS follows.

In the downlink direction, the virtual spacecraft sends imaging, engineering, and science data to the MGDS input data processors. The MGDS receives and processes the telemetry data for display and analysis by users in real time. The received data are cataloged and stored on line for current users or archived off line for future project-specific users.

The uplink capabilities of the MGDS include generation of spacecraft commands -- both real-time and sequences --- and their transmission to the virtual spacecraft. MGDS baseline capabilities include data frame synchronization, data decommutation, data storage and retrieval, and data dissemination. Project-unique requirements may necessitate changes to frame identifications (IDs), commutation structure, channel lists, spacecraft configuration sequence constraints, specific types of data displays, and other parameters as needed. The baseline MGDS workstation software can be adapted to a unique project, thus enabling it to support multiple project-specific space flight operations.

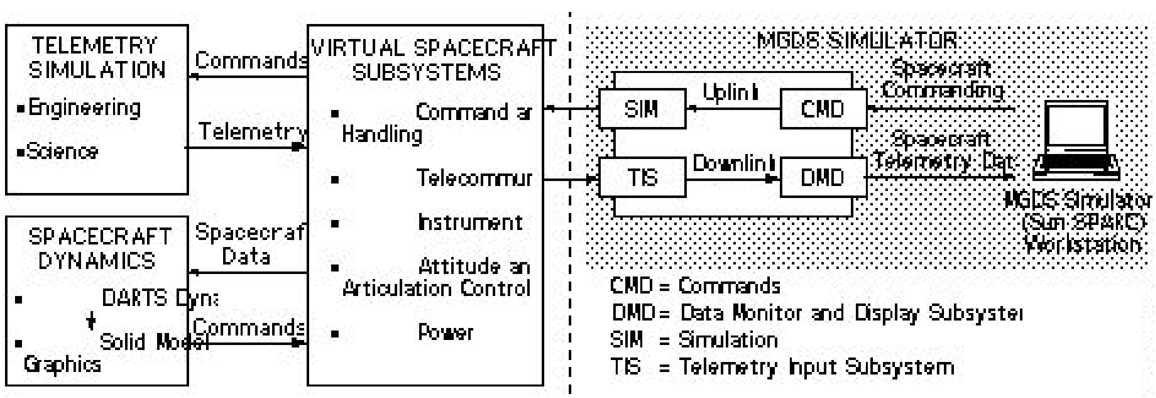

Figure 18. FST Test Configuration.

3.9 Test Operations Console

The Test Operations Console (TOC) captures the configuration and control the operation of the spacecraft subsystem simulations, support equipment, and the FST support systems during the testing and evaluation of a specific EEMOS architecture. This approach provides a consistent and repeatable test environment for spacecraft systems under test and makes maximum use of FST assets. Flight project configurations of hardware and software subsystems and components are controlled within the FST.

Placing a flight project spacecraft subsystem or an entire system into the FST simulation environment as active components of a virtual spacecraft improves confidence in component or spacecraft performance. This closed-loop simulation environment, using a workstation-resident MGDS, provides an uplink and downlink process for command and sequencing data acquisition. A typical test configuration is illustrated in Figure 18.

4

Acronyms

A

AACS	Attitude and Articulation Control Subsystem
A/D	Analog-to-Digital
AMMOS	Advanced Multimission Operations System

C

C&DH	Command and Data Handling
CCSDS	Consultative Committee for Space Data Systems
CDU	Command Detection Unit
CLTU	Command Link Transmission Unit
CMD	Command
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit

D

DAAS	Data Acquisition, Analysis, and Simulation
DARTS	Dynamics Algorithm for Real-Time Simulation
DMD	Data Monitor and Display Subsystem
DMG	Dynamics Model Generator
DSN	Deep Space Network
DST	Deep Space Transponder
DVE	Dynamics Visualization Environment

E

EEMOS	End-to-End Mission Operations System

F

FPGA	Field Programmable Gate Array
FPO	Flight Projects Office
FST	Flight System Testbed

G

GDS	Ground Data System

I

ID	Identification
I/F	Interface
IMOS	Integrated Modeling Optical System
I/O	Input/Output

L

LAN	Local Area Network

M

MGDS	Multimission Ground Data System
MOS	Mission Operations System
MOSO	Multimission Operations Systems Office

N

NMF	New Millennium Flight

P

PC	Personal Computer
PDC	Project Design Center
PDU	Power Distribution Unit
PSC	Power System Controller
Pt-Pt	Point-to-Point
PTM	Proof Test Model
PWR	Power

R

REU	Remote Engineering Unit
RF	Radio Frequency
RFS	Radio Frequency Simulator
RTG	Radioisotope Thermoelectric Generator

S

SBC	Single-Board Computer
SCSI	Small Computer System Interface

S/C	Spacecraft
SDS	Spacecraft Dynamics Simulator
SIM	Simulation
SSR	Solid-State Recorder

T

TCP/IP	Transmission Control Protocol/Internet Protocol
Telecom	Telecommunications
TIS	Telemetry Input Subsystem
TMU	Telemetry Modulation Unit
TOC	Test Operations Console

V

VME/VXI	VersaModule Eurocard/VME Extended